

# Voorstel voor omgaan met “Geen waarde”

## 1 Inleiding

Een element in de inhoud van een bericht kan in sommige gevallen geen waarde hebben. Het niet hebben van een waarde wil echter nog niet zeggen dat er geen betekenis gehecht kan worden aan dit element. Zo kan de betekenis van het niet hebben van de waarde overlijdensdatum bij een persoon betekenen dat deze persoon leeft. Binnen de drie standaarden zijn er verschillende manieren om hier mee om te gaan en wordt dit op verschillend detailniveau gestandaardiseerd. In dit document geven we eerst een feitelijke uitleg van de verschillen. Na een analyse van de verschillen en overeenkomsten geven we tot slot een voorstel voor harmonisatie.

## 2 Analyse

### 2.1 SuwiML/SGR

SuwiML kent de simpelste vorm van standaardisatie. Deze wordt in de SuwiML standaard verwoord volgens onderstaande tabel.

| SuwiML berichtschemata   | SuwiML berichtinstantie waarde beschikbaar   | SuwiML berichtinstantie geen waarde beschikbaar   |
|--------------------------|--|---|
| <i>veld is optioneel</i> | In het geval een veld optioneel is volgens het SuwiML berichtschemata en er is een waarde beschikbaar, dan <u>moet</u> dit veld in het bericht <u>opgenomen worden met</u> een <u>geldige waarde</u> . Het veld <u>mag</u> dan dus <u>niet achterwege blijven</u> . NB. een ongeldige waarde zal vanzelfsprekend tot een foutmelding leiden bij de samenstelling en validatie van het bericht. | In het geval een veld optioneel is volgens het SuwiML berichtschemata en er is geen waarde beschikbaar, dan <u>moet</u> dit veld in het bericht <u>achterwege blijven</u> . Het veld <u>mag</u> dan dus <u>niet</u> met een open- en close-tag <u>worden opgenomen zonder feitelijke waarde</u> .   |
| <i>veld is verplicht</i> | In het geval een veld verplicht is volgens het SuwiML berichtschemata, dan <u>moet</u> dit veld altijd in het bericht <u>opgenomen worden met</u> een <u>geldige waarde</u> . Het veld <u>mag</u> dan dus <u>in geen geval achterwege blijven</u> . NB. een ongeldige waarde zal vanzelfsprekend tot een foutmelding leiden bij de samenstelling en validatie van het bericht.                 | In het geval een veld verplicht is volgens het SuwiML berichtschemata, dan <u>moet</u> dit veld altijd in het bericht <u>opgenomen worden met</u> een <u>geldige waarde</u> . Het veld <u>mag</u> dan dus <u>in geen geval achterwege blijven</u> . NB. het ontbreken (niet beschikbaar zijn) van een geldige waarde zal vanzelfsprekend tot een foutmelding leiden bij de samenstelling en validatie van het bericht. NB. als voor dit veld in het SGR een waarde is gedefinieerd die aangeeft dat het veld leeg/onbekend is, dan dient bij het ontbreken (niet beschikbaar zijn) van een geldige waarde, deze indicatie "leeg/onbekend" te worden gebruikt in het bericht, bijvoorbeeld bij "Geslacht" de waarde "0" indien "onbekend". |

Tabel 5: beperkingen tav het gebruik van lege velden / waarden in een SuwiML bericht.

Kort samengevat komt het er op neer dat als er geen waarde beschikbaar is bij de verzender en het veld optioneel is, het veld weggelaten moet worden. Is het veld verplicht dan moet in het informatiemodel(SGR) een specificatie zijn

opgenomen voor een (fictieve) waarde die “geen waarde” betekent. Deze fictieve waarde wordt dan in het verplichte veld opgenomen.  
Het omgaan met geen waarde is de verantwoordelijkheid van de informatie modelleur en deze heeft per geval de vrijheid om een passende oplossing te kiezen. Hieronder volgt een voorbeeld van een schema en de uitwerking daarvan in een bericht.

```
<element name="Inkomstenperiode" maxOccurs="unbounded">
  <complexType>
    <sequence>
      <element name="DatBlkp" type="sml:Datum"/>
      <element name="DatElkp" type="sml:Datum" minOccurs="0"/>
      <element name="CdSrtlkv" type="sml:CdSrtInkomstenverhouding" minOccurs="0"/>
      <element name="CdTypeArbeidscontract" type="sml:CdTypeArbeidscontract" minOccurs="0"/>
      <element name="CdInvloedVerzekeringsplicht" type="sml:CdInvloedVerzekeringsplicht" minOccurs="0"/>
      <element name="IndLoonheffingskortingToegepast" type="sml:StdIndJN" minOccurs="0"/>
      <element name="IndRegelmatigArbeidspatroon" type="sml:StdIndJN" minOccurs="0"/>
      <element name="IndPersoneelsleningNietInLoon" type="sml:StdIndJN" minOccurs="0"/>
      <element name="IndLoonIsMedeAowAlleenstaande" type="sml:StdIndJN" minOccurs="0"/>
      <element name="IndLoonInclusiefWajongUitkering" type="sml:StdIndJN" minOccurs="0"/>
      <element name="IndVakantiebonnenToegepast" type="sml:StdIndJN" minOccurs="0"/>
      <element name="CdAardlkv" type="sml:CdAardlkv" minOccurs="0"/>
      <element name="CdIncidInkomstenvermindering" type="sml:CdIncidInkomstenvermindering" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
```

Figuur 1 SuwiML Schema voorbeeld

De ellipsen geven aan waar gegevens die optioneel zijn volgens schema, maar niet geleverd worden.

Als er voor een koppelveld gegevens zijn waarvoor een partij niet geautoriseerd is, worden deze gegevens niet opgenomen in het schema.

```
<Inkomstenperiode>
  <DatBlkp>00011030</DatBlkp>
  <DatElkp>00011030</DatElkp>
  <CdSrtlkv>00</CdSrtlkv>
  <IndLoonheffingskortingToegepast>1</IndLoonheffingskortingToegepast>
  <IndRegelmatigArbeidspatroon>0</IndRegelmatigArbeidspatroon>
  <IndPersoneelsleningNietInLoon>0</IndPersoneelsleningNietInLoon>
  <IndLoonIsMedeAowAlleenstaande>0</IndLoonIsMedeAowAlleenstaande>
  <IndLoonInclusiefWajongUitkering>0</IndLoonInclusiefWajongUitkering>
  <IndVakantiebonnenToegepast>1</IndVakantiebonnenToegepast>
  <CdAardlkv>01</CdAardlkv>
</Inkomstenperiode>
```

Figuur 2 SuwiML Bericht voorbeeld

## 2.2 StUF

StUF kent een hoge mate van differentiatie van het begrip “geen waarde”. Het biedt middels het attribuut “StUF:noValue” de mogelijkheid om aan te geven waarom een element geen waarde heeft. Bijvoorbeeld:

```
<ns:object StUF:entiteittype="Persoon">
  <ns:bsn>123456789</ns:bsn>
  <ns:overlijdensdatum xsi:nil="true" StUF:noValue="geenWaarde"/>
</ns:object>
```

In dit geval is de persoon in kwestie nog in leven.

In het volgende geval weten we niet of de persoon wel of niet in leven is omdat de overlijdensdatum onbekend is:

```
<ns:object StUF:entiteittype="Persoon">  
  <ns:bsn>123456789</ns:bsn>  
  <ns:overlijdensdatum xsi:nil="true" StUF:noValue="waardeOnbekend"/>  
</ns:object>
```

Wanneer een element is weggelaten gaan we ervan uit dat de waarde onbekend is. Het onderstaande StUF-fragment is equivalent aan het vorige fragment.

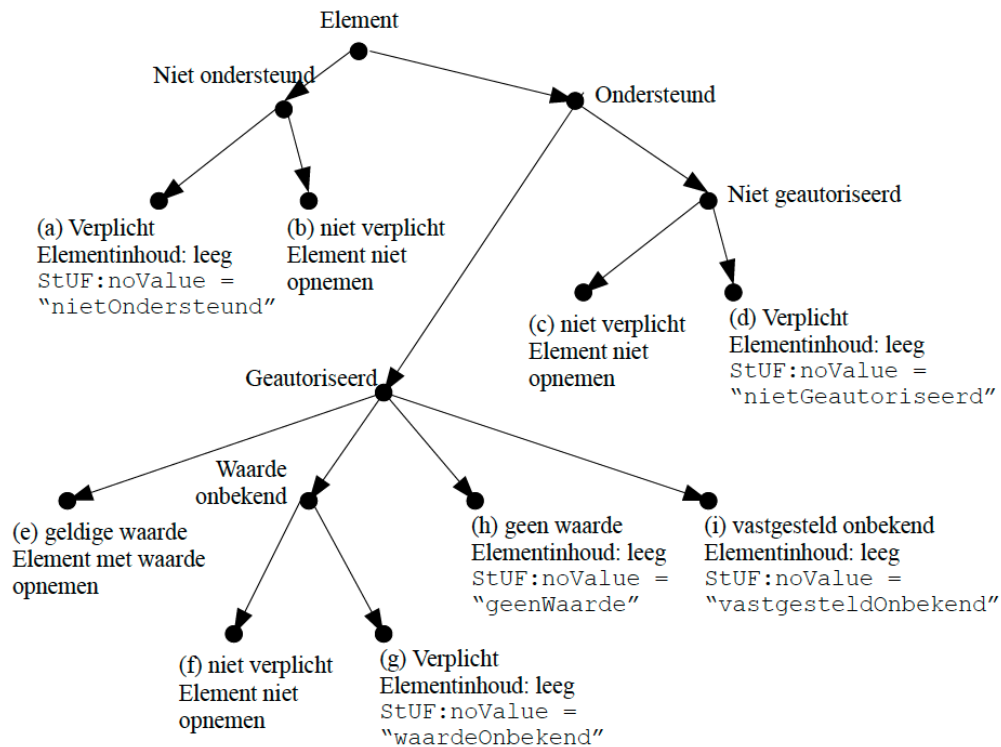
```
<ns:object StUF:entiteittype="Persoon">  
  <ns:bsn>123456789</ns:bsn>  
</ns:object>
```

NB: wanneer een element “geenWaarde” heeft –bij de zender is dus bekend dat er geen waarde is voor dit attribuut- dan mag het element niet worden weggelaten en moet het element altijd worden opgenomen in het StUF-bericht.

Hieronder de definitie van het attribuut “noValue” in XML schema:

```
<attribute name="noValue" type="StUF:NoValue"/>  
<simpleType name="NoValue">  
  <restriction base="string">  
    <enumeration value="nietOndersteund"/>  
    <enumeration value="nietGeautoriseerd"/>  
    <enumeration value="geenWaarde"/>  
    <enumeration value="waardeOnbekend"/>  
    <enumeration value="vastgesteldOnbekend"/>  
  </restriction>  
</simpleType>
```

StUF kent een aantal criteria op basis waarvan een andere vorm van “geen waarde” geleverd kan worden. Dit wordt in de standaard geïllustreerd met onderstaande boom.



Ten eerste is er de vraag of de zender die de gegevens levert het gevraagde element ondersteund (d.w.z. opgenomen heeft in zijn administratie). Met StUF is het mogelijk een service te implementeren die niet alle verplichte elementen uit een bijbehorende XSD ondersteund. Wordt er om zo'n element gevraagd, dan wordt "nietOndersteund" teruggegeven.

StUF kent ook autorisaties, met name bij persoonsgegevens komen deze veel voor. Indien een vrager geen autorisatie heeft, kan hij "nietGeautoriseerd" als waarde terugkrijgen.

Indien autorisatie en ondersteuning geen rol spelen zijn er overige gevallen voor "geen waarde". Te beginnen bij waardeOnbekend, het systeem heeft geen waarde maar weet wel dat er een waarde moet bestaan.

Voor alle tot nu toe beschreven gevallen geldt dat indien het element niet verplicht is het niet opgenomen wordt.

Tot slot kent StUF twee gevallen waarbij geenwaarde altijd opgenomen wordt of het element optioneel is of niet. Ten eerste "geenWaarde" het element heeft nog geen waarde en dit heeft betekenis, bijvoorbeeld het overlijdensdatum voorbeeld uit de inleiding. Tot slot "vastgesteldOnbekend" een element zou een waarde moeten hebben maar het is vastgesteld dat deze nooit meer te achterhalen is. Bijvoorbeeld wanneer het brondocument verloren is gegaan.

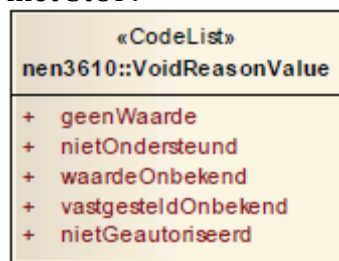
Bij deze laatste twee wordt StUF:noValue altijd opgenomen aangezien de betekenis subtiel verschilt. Bij de eerste drie houdt geen waarde in: er moet nu een waarde zijn maar door omstandigheden aan server of client kant krijg je die niet. Bij (h) geldt: er is op dit moment geen geldige waarde in te vullen anders dan geen waarde. Bij (i) geldt: het is onmogelijk om de waarde op dit moment op wat voor manier dan ook te achterhalen.

### 2.3 NEN3610

In de NEN3610 wordt “geen waarde” in UML aangegeven met het stereotype <<voidable>> dat kan worden gebruikt op attributen en associaties. Hier een voorbeeld van een UML klasse met een attribuut ‘naam’ waar dit is toegepast:



Met daarbij een lijstje mogelijke redenen van de ontbrekende waarde, afgestemd met StUF:



In de NEN3610<sup>1</sup> is dit als volgt gedefinieerd:

---

<sup>1</sup> <http://www.nen.nl/web/Normshop/Norm/NEN-36102011-nl.htm>

|                    |  |
|--------------------|--|
| Klasse: «codeList» | VoidReasonValue                            |
| Alias              | Void reason value                          |
| Definitie          | reden waarom een 'void-waarde' is ingevuld |
| Herkomst definitie | NEN 3610                                   |
| Stereotype         | CodeList                                   |
| Toelichting        | Zie hoofdstuk 8.                           |

|                             |   |
|-----------------------------|---|
| Waarde: geenWaarde          |   |
| Definitie                   | element heeft in werkelijkheid geen waarde  |
| Waarde: nietOndersteund     |   |
| Definitie                   | zender houdt in zijn registratie geen waarde voor dit attribuut bij. Geldt voor alle objecten van dit objecttype  |
| Waarde: waardeOnbekend      |   |
| Definitie                   | element is verplicht maar de waarde is bij de zender niet bekend  |
| Waarde: vastgesteldOnbekend |   |
| Definitie                   | er is vastgesteld dat de waarde van het attribuut onbekend is en hoogst waarschijnlijk niet meer kan worden achterhaald (bijvoorbeeld omdat het brondocument onleesbaar is of het object niet meer bestaat in de werkelijkheid) |
| Waarde: nietGeautoriseerd   |   |
| Definitie                   | deze zender vindt dat de ontvanger niet geautoriseerd is om de waarde te kennen   |

| NEN 3610 VoidReasonValue | Inspire VoidReasonValue  |
|--------------------------|--------------------------|
| geenWaarde               | No Value; Not applicable |
| nietOndersteund          | Unpopulated              |
| waardeOnbekend           | Unknown                  |
| vastgesteldOnbekend      | Unknown                  |
| nietGeautoriseerd        |                          |

In een GML application schema ziet dat er zo uit:

```
<complexType name="WegType">
  <sequence>
    <element name="hartlijn" type="gml:CurvePropertyType"/>
    <element name="naam" nillable="true">
      <complexType>
        <simpleContent>
          <extension base="string">
            <attribute name="nilReason" type="gml:NilReasonType"/>
          </extension>
        </simpleContent>
      </complexType>
    </element>
  </sequence>
</complexType>
```

En in een GML data bestand:

```
<Weg>
  <hartlijn><gml:Curve>...</gml:Curve></hartlijn>
  <naam xsi:nil="true" nilReason="geenWaarde"/>
</Weg>
```

## 2.4 Overeenkomsten

In alle gevallen lijkt het te gaan om een situatie waar een vragende partij gegevens vraagt van een leverende partij die dit (mogelijk) al in een administratie heeft opgenomen. Het lijkt niet te gaan om een situatie waarbij gegevens die bijvoorbeeld ingetikt of berekend zijn in de zendende applicatie en worden doorgegeven aan de ontvangende applicatie om geregistreerd te worden.

In alle gevallen gaat het om situatie waarin er geen sprake is van een technische fout, maar van een functioneel herkenbare situatie,

In alle gevallen wordt de situatie onderkend dat er in de administratie een leeg veld is (technisch gezien een null value of een fictieve waarde (bijv. 99-99-9999 in een datumveld). Alleen in de StUF en NEN3610 situatie wordt daar echter bij onderkend dat dit meerdere dingen kan betekenen:

- De waarde bestaat niet (overlijdensdatum van een levend persoon)
- De waarde is niet bekend (geboortedatum van een persoon die dit zelf ook niet weet of waar we de vraag niet hebben gesteld)
- De waarde is niet bekend na goed onderzoek (geboortedatum van een immigrant die het zelf niet weet en waarbij er geen enkel formeel document is gevonden waar het op is vastgelegd).

Tussen de standaarden bestaan veel overeenkomsten. StUF en NEN3610 hebben op dit onderwerp al in het verleden geharmoniseerd en onderkennen dezelfde waardelijst voor met mogelijkheden voor geenWaarde. StUF en SuwiML hanteren een zeer vergelijkbare methode daar waar het omgang met optionele velden/elementen betreft. Binnen SuwiML vindt de definitie van een veld zonder waarde dat moet worden gecommuniceerd plaats op informatiemodel Niveau in het SGR. NEN3610 onderkent ook op informatiemodel/UML niveau dat een attribuut stereotype "Voidable" is. Bij StUF is dit niet het geval.

## 2.5 Verschillen

In geval van StUF wordt een berichtmodel gespecificeerd onafhankelijk van de situatie waarin die gebruikt gaat worden. Dat leidt tot de mogelijke situaties dat:

- A) Een bericht een verplicht element benoemt, welke niet door de supplier geleverd kan worden.
- B) Een bericht een element benoemt, waar de consumer niet voor geautoriseerd is.

Het lijkt er op dat dit in de SuwiML context niet van toepassing is.

NEN3610 en StUF zijn al geharmoniseerd, maar er zijn nog wel verschillen.

- De NEN 3610 kent naast de StUF waarde lijst voor geen waarde ook een lijst van Inspire met andere waarden. Deze worden toegepast bij Inspire plichtige datasets.
- De XML is bij alle standaarden duidelijk anders van vorm.
- Daarnaast onderkent StUF een aantal gevallen waarin "geenWaarde" altijd wordt doorgegeven ook als het element optioneel is, terwijl bij



SuwiML de regel is om bij geen waarde optionele velden altijd weg te laten

Dit leidt in de praktijk tot verschillende manieren voor het omgaan met geen waarde: in SuwiML wordt de overlijdensdatum weggelaten indien een persoon nog leeft, bij StUF kan de overlijdensdatum weggelaten worden indien niet bekend is of een persoon nog leeft. Is zeker dat een persoon nog leeft dan dient StUF:noValue = geenWaarde te worden teruggegeven. Dit kan leiden tot miscommunicatie.

### 3 Voorstel tot harmonisatie

De mogelijkheid tot harmonisatie is hier zeer groot. Het voorstel is om de volgende vier punten door te voeren:

1. Opnemen van de codelijst in alle standaarden  
Het laagst hangend fruit is de code lijst die StUF en NEN3610 al hanteren. Deze zou ook bij modellering in SGR overgenomen kunnen worden voor het meegeven van “geen waarde” gevallen bij verplichte velden. Daarmee komt deze codelijst ook in SuwiML verkeer terecht.  
De NEN3610, StUF en SuwiML gebruiken dezelfde codelijst en bijbehorende definities voor “geen waarde” en modelleren deze vanuit hun informatiemodellen

|                             |   |
|-----------------------------|---|
| Waarde: geenWaarde          |   |
| Definitie                   | element heeft in werkelijkheid geen waarde  |
| Waarde: nietOndersteund     |   |
| Definitie                   | zender houdt in zijn registratie geen waarde voor dit attribuut bij. Geldt voor alle objecten van dit objecttype  |
| Waarde: waardeOnbekend      |   |
| Definitie                   | element is verplicht maar de waarde is bij de zender niet bekend  |
| Waarde: vastgesteldOnbekend |   |
| Definitie                   | er is vastgesteld dat de waarde van het attribuut onbekend is en hoogst waarschijnlijk niet meer kan worden achterhaald (bijvoorbeeld omdat het brondocument onleesbaar is of het object niet meer bestaat in de werkelijkheid) |
| Waarde: nietGeautoriseerd   |   |
| Definitie                   | deze zender vindt dat de ontvanger niet geautoriseerd is om de waarde te kennen   |

Omdat er geen 100% zekerheid is of dit alle mogelijke relevante waarden zijn, is het voorstel om deze codelijst als “uitbreidbare” lijst op te nemen in de standaard. Dat betekent dat als je iets wilt communiceren over geen waarden dat je het dan op deze manier doet, behalve als je een betekenis wilt overdragen die hiermee niet weer te geven is. In dat laatste geval is het uiteraard goed om dat ook als RFC weer aan te dragen zodat de lijst met opgenomen codes zo volledig mogelijk is.

2. Modelleren van “geen waarde” op informatiemodel niveau.  
Zowel SuwiML (of eigenlijk SGR) en NEN3610 modelleren “geen waarde” ook op informatiemodel niveau, StUF en bijbehorende informatiemodellen (RSGb, RGBZ, ZTC etc.) kunnen dit ook overnemen immers “geen waarde” is een onderwerp waar semantiek in verscholen zit. Dat zou ook op informatiemodel niveau onderkend moeten worden.



Let op:

In bestaande administraties kan de betekenis van lege velden ambivalent zijn. Het kan zowel geenWaarde, waardeOnbekend als vastgesteldGeenWaarde betekenen. Wil je voorkomen dat je in deze gevallen eerst heel veel administratief werk moet doen om de ambivalentie weg te werken, dan moeten de berichten die dit communiceren deze ambivalentie aankunnen. (M.a.w. het attribuut "StUF:novalue" moet dus optioneel zijn in de berichten). Bovendien zal een administratie waar deze ambivalentie is weggewerkt geen gegevens meer geautomatiseerd kunnen opnemen van een ambivalente administratie: eerst moet n.l. de ambivalentie worden opgelost.

3. Op één manier berichten samenstellen als er geen waarde is.  
Voor de gevallen waarin StUF ook bij optionele velden "geen waarde" doorgeeft komt een best practice waarmee informatiemodellereurs eenduidig deze waarden kunnen toepassen. Deze harmonisatie kan op korte termijn gerealiseerd worden.
4. Betekenis van het weglaten van elementen gelijktrekken.  
Ten eerste betekent dit een keuze maken:  
Kiezen we vanaf een nog te definiëren moment voor: a) weglaten is gelijk aan waardeOnbekend of b) weglaten is gelijk aan geenWaarde of c) opnemen van een parameter in een bericht welke keuze in dat bericht is gemaakt.  
Ten tweede betekent het dat voor de bestaande berichtspecificaties bepaald moet worden welke interpretatie geldt. Voor de bestaande koppelingen zul je een migratiestrategie moeten gaan bedenken.

Volledige harmonisatie, precies dezelfde XML bij alle standaarden is lastiger. GML is een internationale standaard daar is dus weinig tot geen bewegingsruimte. Daar zal allereerst dit punt moeten worden geadresseerd. Vervolgens is het afwachten of deze ook wordt doorgevoerd. In StUF is StUF:novalue geïntegreerd in de StUF methodiek.

Voor de langere termijn kan er gekeken worden naar een eenduidige XML structuur voor de situaties waarbij geen waarde wordt geleverd. Een dergelijke harmonisatie in XML structuur kan echter beter opgepakt worden in een bredere context waarbij de structuur van meer zaken dan alleen geen waarde geharmoniseerd wordt.